



# Mississippi State UNIVERSITY

## Center for Air Sea Technology

### FY98 STUDENT RESEARCH PROJECTS

Technical Note 2-98  
30 September 1998

19981102 119

This research was supported by the Department of the Navy, Office of the Chief of Naval Research (ONR) under Research Grants N00014-95-1-0203, N00014-97-1-0099, and N00014-97-1-0525 with MSU; Naval Research Laboratory (NRL) under Contracts N00014-95-C-6032 and N00173-98-C-6012; NASA Contract NAS13-564 Delivery Orders 131 and 132 with the Naval Oceanographic Office; and Contract MXS984100-97-0 with the Environmental Protection Agency.

DTIC QUALITY INSPECTED 4

Approved for public release; distribution is unlimited.  
Mississippi State University Center for Air Sea Technology  
Stennis Space Center, MS 39529-6000

**MISSISSIPPI STATE UNIVERSITY  
CENTER FOR AIR SEA TECHNOLOGY**

**FY98  
STUDENT RESEARCH PROJECTS**

BUILDING 1103-ROOM 233

STENNIS SPACE CENTER, MS 39529-6000

Compiled and Edited by  
Lanny A. Yeske

30 September 1998

This research was supported by the Department of the Navy, Office of the Chief of Naval Research (ONR) under Research Grants N00014-95-1-0203, N00014-97-1-0099, and N00014-97-1-0525 with MSU; Naval Research Laboratory (NRL) under Contracts N00014-95-C-6032 and N00173-98-C-6012; NASA Contract NAS13-564 Delivery Orders 131 and 132 with the Naval Oceanographic Office; and Contract MXS984100-97-0 with the Environmental Protection Agency. The opinions, findings, conclusions, and recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the U.S. Government. No official endorsement should be inferred.

## FOREWARD

The Mississippi State University Center for Air Sea Technology (CAST) research program in FY98 included ONR grants in the areas of High Resolution Coastal Modeling of the Santa Barbara Channel, Relocatable Model Studies in Coastal Regions, and Modeling in the North Atlantic with Data Assimilation. The research program also encompassed several other contracts including the Navy Interactive Data Analysis System (NIDAS) with NAVOCEANO and Commander Mine Warfare Command, the Master Environmental Library with the Naval Research Laboratory, and the EPA Gulf of Mexico Program for Software Infrastructure for Integrated Acquisitions and Manipulation of Environmental Data.

Our objectives are closely focused on operational requirements and include the following:

1. To develop advanced tools and methodologies for model evaluation, validation and visualization, both oceanographic and atmospheric.
2. To develop a system-level capability for conducting temporally and spatially scaled ocean simulations that are driven by or are responsive to ocean models, and incorporates the coupling to atmospheric models.
3. To develop an oceanographic and atmospheric data management system with emphasis on distributed databases in a network environment, and with database optimization and standardization, including the use of the World Wide Web (WWW).
4. To implement a high performance parallel computing technology for CAST ocean models.
5. To implement new software applications and systems into a distributed object computing environment utilizing government and industry standards such as CORBA (Common Object Request Broker Architecture), and the Extensible Distributed Graphical Environment (EDGE).

In accomplishing these objectives, CAST in FY1998 supported 10 graduate and undergraduate students, which included one through the MSU Cooperative Education Program, seven from the University of Southern Mississippi, one from Mississippi Gulf Coast Community College, and one from Florida State University.

This technical note summarizes the FY1998 research conducted by these students. CAST was extremely pleased with the research support provided by these individuals, not only in their dedication but also in the quality of the research conducted.

  
LANNY A. YESKE  
Deputy Director

## TABLE OF CONTENTS

FOREWARD .....	i
Validation of the CURReNTSS (University of Colorado Relocatable Nested Tide Storm Surge) Model and CURReNTSS Model Transition by Mr. Shuvobroto Brahmachari, University of Southern Mississippi and Mr. Steve W. Payne, Mississippi State University .....	1
Extraction of Levitus Climatological Data for use in the Relocatable Modeling Environment (RME) Project by Ms. Patricia Ehrensing, University of Southern Mississippi and Mr. Shuvobroto Brahmachari, Mississippi State University .....	3
Design and Implementation of a CAST Book Inventory System by Mr. Sayantan Ganguly, University of Southern Mississippi and Mr. Steve W. Payne, Mississippi State University .....	4
Installing Script for the Relocatable Modeling Environment (RME) by Mr. Sayantan Ganguly, University of Southern Mississippi and Mr. Steve W. Payne, Mississippi State University .....	6
CAST System Analysis by Mr. Sayantan Ganguly, University of Southern Mississippi, and Mr. Steve W. Payne, Mississippi State University .....	6
Record Management tests for the Master Environmental Library (MEL) Project by Mr. Sayantan Ganguly, University of Southern Mississippi and Mr. Ming Zhou, Mississippi State University .....	9
Conversion Application for use with the Relocatable Modeling Environment (RME) by Mr. Brandon Jockman and Mr. Shuvobroto Brahmachari, Mississippi State University .....	10
World Wide Web Development utilizing Hypertext Markup Language (HMTL), Common Gateway Interface (CGI), and Practical Extraction and Report Language (PERL) by Mr. Brandon Jockman and Mr. Steve W. Payne, Mississippi State University .....	10
Design of Communications Between Multiple Programming Language's for the Master Environmental Library (MEL) by Mr. Brandon Jockman and Mr. John Chambless, Mississippi State University .....	11
Analysis of XMT Technology by Mr. Brandon Jockman and Mr. Ming Zhou, Mississippi State University .....	12

Embeddable Generic Object Processing Dynamic Tree for Graphical User Interfaces (GUI) by Mr. Brandon Jockman and Mr. Ming Zhou, Mississippi State University .....	13
Alpha Testing and Code Revisions for the Master Environmental Library (MEL) Resource Site Software (MRSS) by Mr. Brandon Jockman, Mississippi State University, Mr. Sayantan Ganguly, University of Southern Mississippi, Mr. Sachin Bhate and Mr. John Chambless, Mississippi State University.....	17
A Three-Dimensional Scalar Field Initialization Program for the Baroclinic Tidal Model in the Relocatable Modeling Environment (RME) Project by Mr. Colin Murray, Florida State University, and Mr. Shuvobroto Brahmachari, Mississippi State University .....	18
Research on the Oracle8 Database Management System and Web Server System Configuration by Mr. Shengyong Li, University of Southern Mississippi, and Mr. Ming Zhou, Mississippi State University .....	20
Development of an Extractor for the Master Environmental Library (MEL) by Mr. Shengyong Li, University of Southern Mississippi, and Mr. Steve Foster, Mississippi State University .....	20
Development of an Extraction Library Module for the Master Environmental Library (MEL) by Mr. Shengyong Li, University of Southern Mississippi, and Mr. Steve Foster, Mississippi State University .....	21
Development of a Virtual Data Warehouse (VDW) for the Environmental Protection Agency/Gulf of Mexico Program Office (GMPO) by Mr. Mike Zhou, University of Southern Mississippi, and Mr. Ming Zhou, Mississippi State University .....	22
Development of the Enterprise Information Explorer by Mr. Mike Zhou, University of Southern Mississippi, and Mr. Ming Zhou, Mississippi State University .....	22
Inventory CAST Equipment and Apply New MSU Inventory Requirements, Learn the Macintosh Operating System and Troubleshoot Problems, Purchasing Procedures, and Design and Development of a backup for Macintosh Computers by Mr. Gabriel Murchison, Mississippi Gulf Coast Community College, and Mr. Steve W. Payne, Mississippi State University .....	24
Enhancement of the Naval Interactive Data Analysis System (NIDAS) in Axes Units by Mr. Rohit Mehra, University of Southern Mississippi, and Mr. Clifton Abbott, Mississippi State University .....	25

Implementing a Data Category Interface by Mr. Rohit Mehra, University of Southern Mississippi, and Mr. Clifton Abbott, Mississippi State University.....	25
Design of a NetCDF File Format for the Comprehensive Oceanographic and Atmospheric Data Set (COADS) by Mr. Rohit Mehra, University of Southern Mississippi, and Mr. Clifton Abbott, Mississippi State University.....	26
The Application of JavaScript Technology to the EPA/GMPO's Virtual Data Warehouse (VDW) Graphical User Interface (GUI) by Mr. Min Zhang, University of Southern Mississippi, and Mr. Ming Zhou, Mississippi State University .....	27
Distribution List.....	29

**Shuvobroto Brahmachari**  
**Ph.D. Program in Scientific Computing**  
**University of Southern Mississippi**

**Project #1 Title:** Validation of the CURReNTSS (University of Colorado Rapidly Relocatable Nested Tide Storm Surge) Model.

**Objectives:** Under ONR Research Grant N00014-95-1-0203 to validate model results with real data.

**Approach:** To make multiple runs of the CURReNTSS model using a combination of several astronomical components to gauge their respective influence on the results. The regions concentrated on were the North Atlantic Bight and the Straits of Juan de Fuca. Electronic records of current meter observations, sea surface elevation, and position data from International Hydrographic Office (IHO) stations in the North Atlantic Bight were created for assimilation into CURReNTSS. The U.S. Geological Survey, National Ocean Survey, and ETOPO-5 data were integrated to generate a high-resolution bathymetry for the North Atlantic Bight. The co-phase and co-amplitude of sea surface height and current velocities were plotted for several CURReNTSS runs with several combinations of astronomical tidal components.

**Results:** A much more detailed bathymetry was obtained for the North Atlantic Bight Region, which resulted in better model runs. The assimilation of the IHO station data in the model also yielded better results.

**Future Research Recommendations:** The integration of bathymetry from different locations around the world would enhance the resolution of the global bathymetry database, and make the model results more reliable. More tidal station data is also needed for better assimilation during model runs.

**Research Advisor:** Mr. Steve W. Payne, Center for Air Sea Technology, Mississippi State University.

.....  
**Project #2 Title:** CURReNTSS Model Transition.

**Objectives:** Transition of the CURReNTSS Model from the Tidal Relocatable Modeling Environment (RME) to the Common Modeling Relocatable Operating Environment (CMROE).

**Approach:** This project involved designing the CMROE to implement model execution and post-processing under the Extensible Distributed Graphics Environment (EDGE) architecture. Since the many steps involved in setting up a relocatable numerical model in a new region of interest are tedious and labor intensive, there is a need for simplification and organization of tasks. Enhancements were necessary to the existing RME to improve its functionality and to integrate the entire system into an object-oriented, user-friendly, interactive graphical user interface (GUI). Specific enhancements included the following: common operating environment,

object-oriented management of data and tools, flexible flow control, automated tasks, and enhanced initialization data.

The specifications for the CMROE software requirements were determined and a project timeline created. The new GUI was completely restructured with little resemblance to the old one. Re-designs included user flexibility and ease of process control, with object-oriented drag and drop features in the EDGE architecture. A post-processor filter was also designed and created to convert the model results to a netCDF format, which could then be read by the Visualizer Module using NCAR Graphics. The Visualizer Module was modified to accommodate the Region Selector. The Region Generator back-end was enhanced to include dynamic memory allocation by altering the model code. Intermediate filters for changing file formats from ASCII to netCDF and vice-versa, were coded for use in the pre-processor.

**Results:** The model result files, when run through the post-processor filter, generated netCDF files which were viewed with the Visualization Engine. The preprocessor code was changed to allocate dynamic memory resulting in more efficient model runs. The redesigned GUI was provided with more features in region selection such as nesting.

**Future Research Recommendations:** The object-oriented approach used in this project will eventually allow it to incorporate any model. Other models, such as DieCAST, should be assimilated into the CMROE to test the versatility of EDGE as a second phase of CMROE development. Future research would examine the versatility of CMROE with 3D models.

**Research Advisor:** Mr. Steve W. Payne, Center for Air Sea Technology, Mississippi State University.



**Patricia M. Ehrensing**  
**B.S. Program in Applied Computer Science**  
**University of Southern Mississippi**

**Project Title:** Extraction of Levitus Climatological Data for use in the Relocatable Modeling Environment (RME) Project.

**Objective:** Develop a routine to extract data from a network Common Data Format (netCDF) file. This routine will improve the efficiency of the FORTRAN subprogram contained in the backend of the RME.

**Approach:** The first step was to investigate the mechanism by which the current code mapped the Levitus climatological temperature and salinity values onto a sigma-coordinate model grid. The subprogram was written in FORTRAN and used a flat file to retrieve the desired data. After determining what input was required for the FORTRAN subroutine, the next step was to create a C function to perform the extraction. The C function extracted the temperature and salinity values from a netCDF file for a given area as specified by the model domain. The extracted temperature and salinity arrays were passed to the FORTRAN subroutine which calculated the sigma levels and returned the new salinity and temperature arrays, by sigma levels, back to the C program.

**Results:** The time needed for the EXTS extraction subroutine to calculate sigma levels was greatly decreased. This increased efficiency was due to the ability of the extraction routine to retrieve only the necessary portion of the Levitus climatology. Originally the entire Levitus climatology was read into memory followed by the extraction.

**Research Advisor:** Mr. Shuvobroto Brahmachari, Center for Air Sea Technology, Mississippi State University.

**Sayantana Ganguly**  
**B.S. Program in Computer Science**  
**University of Southern Mississippi**

**Project #1 Title:** Design and Implementation of a CAST Book Inventory System.

**Background:** The book inventory is a data access system accessible via the World Wide Web. CGI scripts were used to enable access and transfer between the web and the inventory system.

**Objective:** The primary objective was to build an inventory system to enable access to a database. The support for add, delete and search capabilities was essential. There was also a need for concise error detection and strict rules for entering data.

**Approach:** After reviewing system requirements, I first designed a general layout of the system, which is shown in Figure 1.

The '*cgi*' directory was created to handle the addition of books. The *cgi* script was designed to return values from the GUI and process them. If a certain book already existed, then a script to inform the user of this and options to update the quantity field was needed.

The '*cgi-bin*' directory was designed to allow for deletion of books. Upon entering the required fields, the system searches through the inventory and retrieves the book if it exists. There can only be one match since an ISBN number must be entered. On returning the book, the delete option is presented and deleted if activated. Both the adding and deleting of the books are password protected.

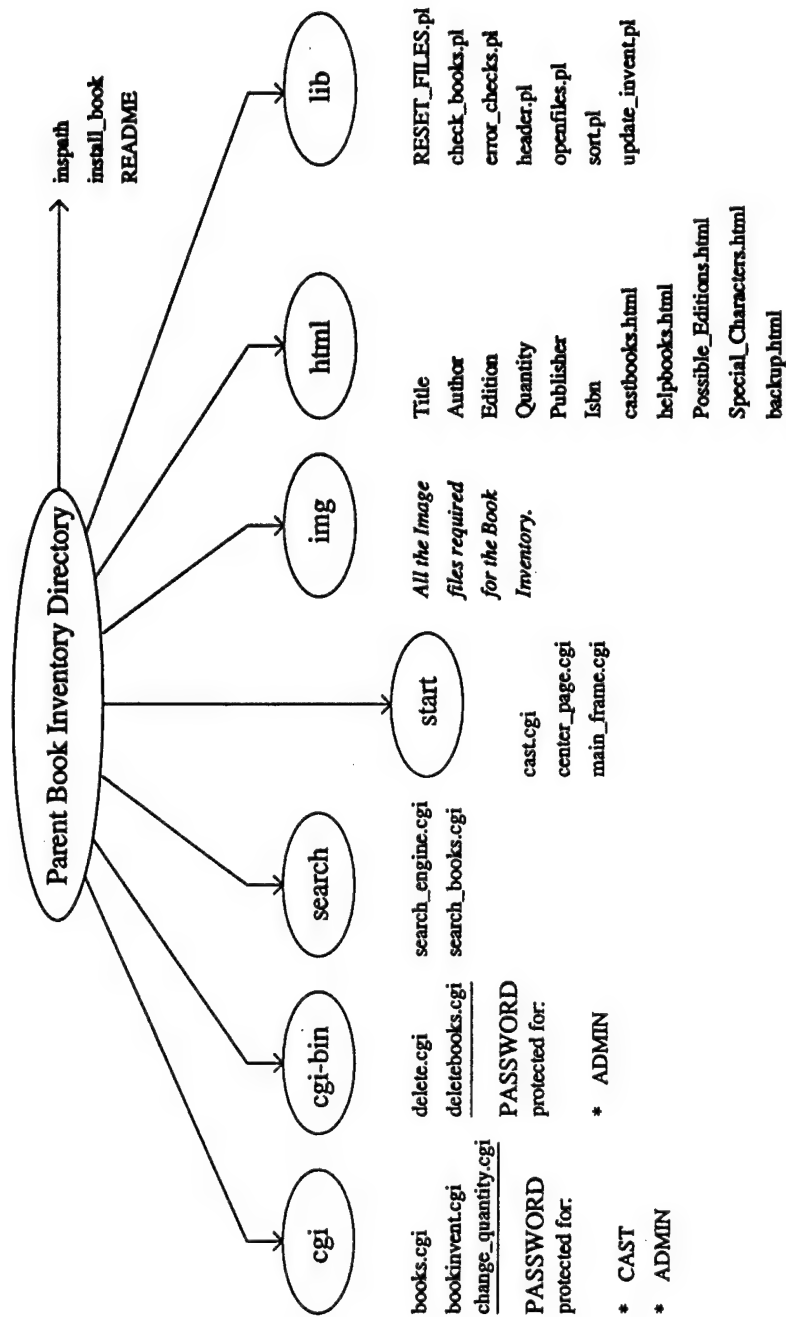
The '*search*' directory maintains the search page, and is accessible by all researchers. This allows the inventory to be searched using two types of patterns: an exact or a partial match. If any fields are known, then the exact match will save user time by returning the books with this desired attribute in it.

The '*html*' directory contains files that serve as the database. It also contains the help and other pages that display error messages. The '*lib*' directory was designed to hold commonly used procedures and functions. The '*img*' directory was created to hold the images/icons used.

Finally, the '*start*' directory holds files that generate the start up page when the system is accessed. It has a target frame where the input and results are displayed. There is also another frame that allows for an easy way to flip between different pages.

**Results:** The resulting system is installed into a directory by running the '*install\_book*' script. The process to install it is in the README file. Once installed, this software can be used to add, delete and search for books. The system also has a description of errors that may occur.

**Research Advisor:** Mr. Steve W. Payne, Center for Air Sea Technology, Mississippi State University.



**FIGURE 1: CAST Book Inventory System**

**Project #2 Title:** Installing Script for the Relocatable Modeling Environment (RME).

**Objective:** To develop a script that facilitates the 'make'ing of directories in the RME Project. Previously, code changes within a directory required the 'make'ing of the directory which generated executable code that must be copied into a designated directory as well as storing library files into specified directories. Modifications of certain directories required a 'makefile' in related directories also to be executed. The main objective of the install script was to improve this approach and to have a script perform these tasks.

**Approach:** The first step was to become familiar with directory links and to understand what executables were generated when a specific directory has executed a makefile and where executables are stored. I also needed to become aware of which directories generated executables and which needed libraries. Once compiled, an efficient algorithm was to be generated to minimize repetition of code and enhance the speed of execution.

**Results:** The install script now enables the movement of files by providing the directory modified as an argument passes into it. There is also a default argument of 'all' that presumes all the directories were modified, so it 'make's all the directories and copies whatever needs to be.

**Research Advisor:** Ms. Yee Lau, Center for Air Sea Technology, Mississippi State University.

.....

**Project #3:** CAST System Analysis.

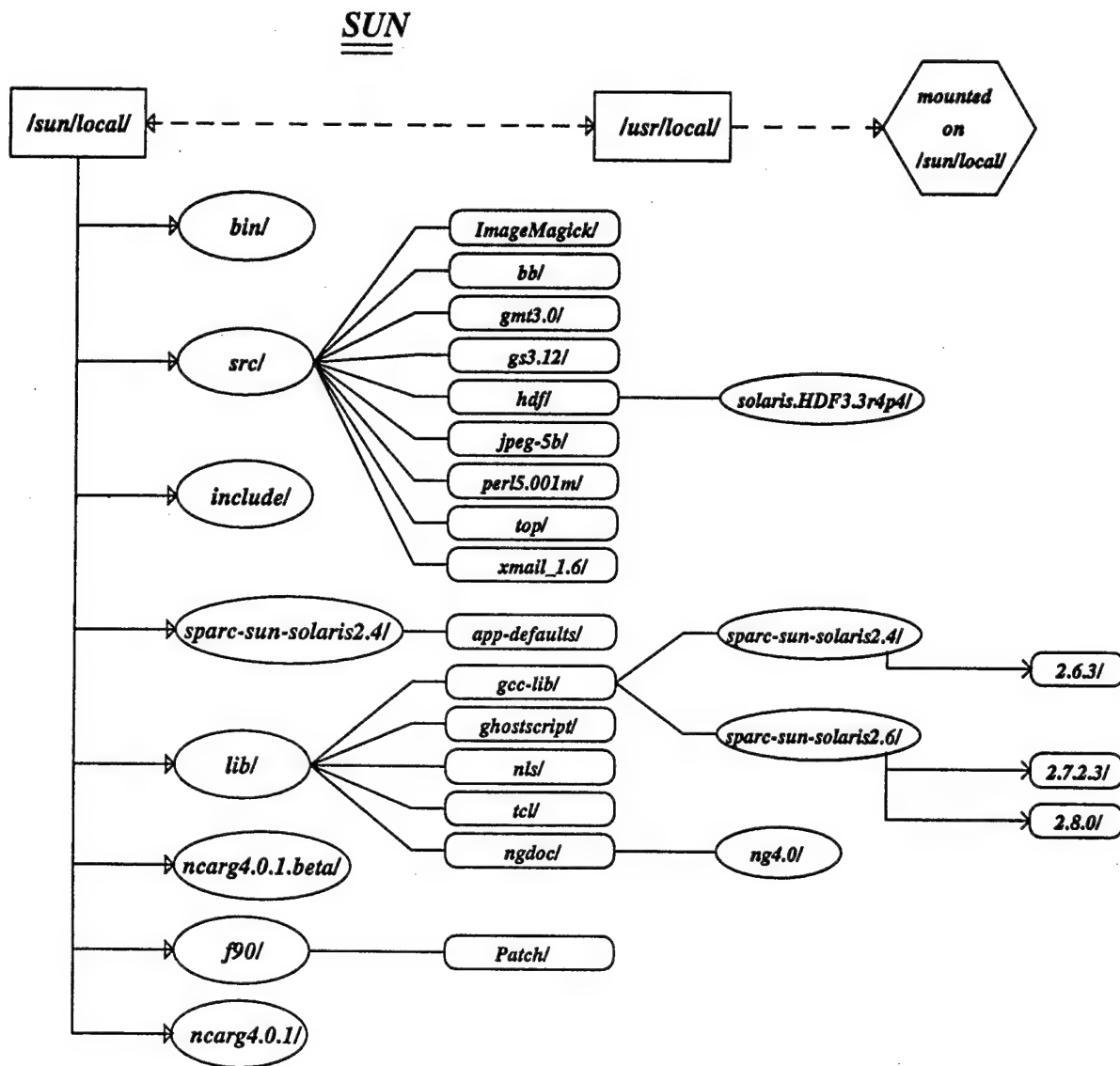
**Objective:** To generate a report of all software and independent executables accessible by users at CAST in the SUN, SGI, and data/cast/local directories. Also to be included, was the /usr/local directory in each of the SUN and SGI machines. This report was to serve as a guide to what currently exists in the system.

**Approach:** The systematic analysis of these directories was begun on a SUN machine. A 'tree' program was used to automatically generate a diagram that displays the subdirectories. This was then used to check each directory and file for version numbers and categorically group them by software. Any files not executing or generating errors were noted for removal.

**Results:** The final report contained the working executables and software present in the SUN and SGI machines. The list of files grouped by software enables the administrator to see files currently working and associate them with a particular software. These are shown in Figures 2 and 3.

**Future Research Recommendations:** Conduct work in the /data/cast/local/sun/bin directory and provide a list of applicable subdirectories. There are also numerous 'tar' and other compressed files that exist, even after the software is extracted. Removal of compressions of versions no longer used is advisable.

**Research Advisor:** Mr. Steve W. Payne, Center for Air Sea Technology, Mississippi State University.



**FIGURE 2: SUN Files**

SGI

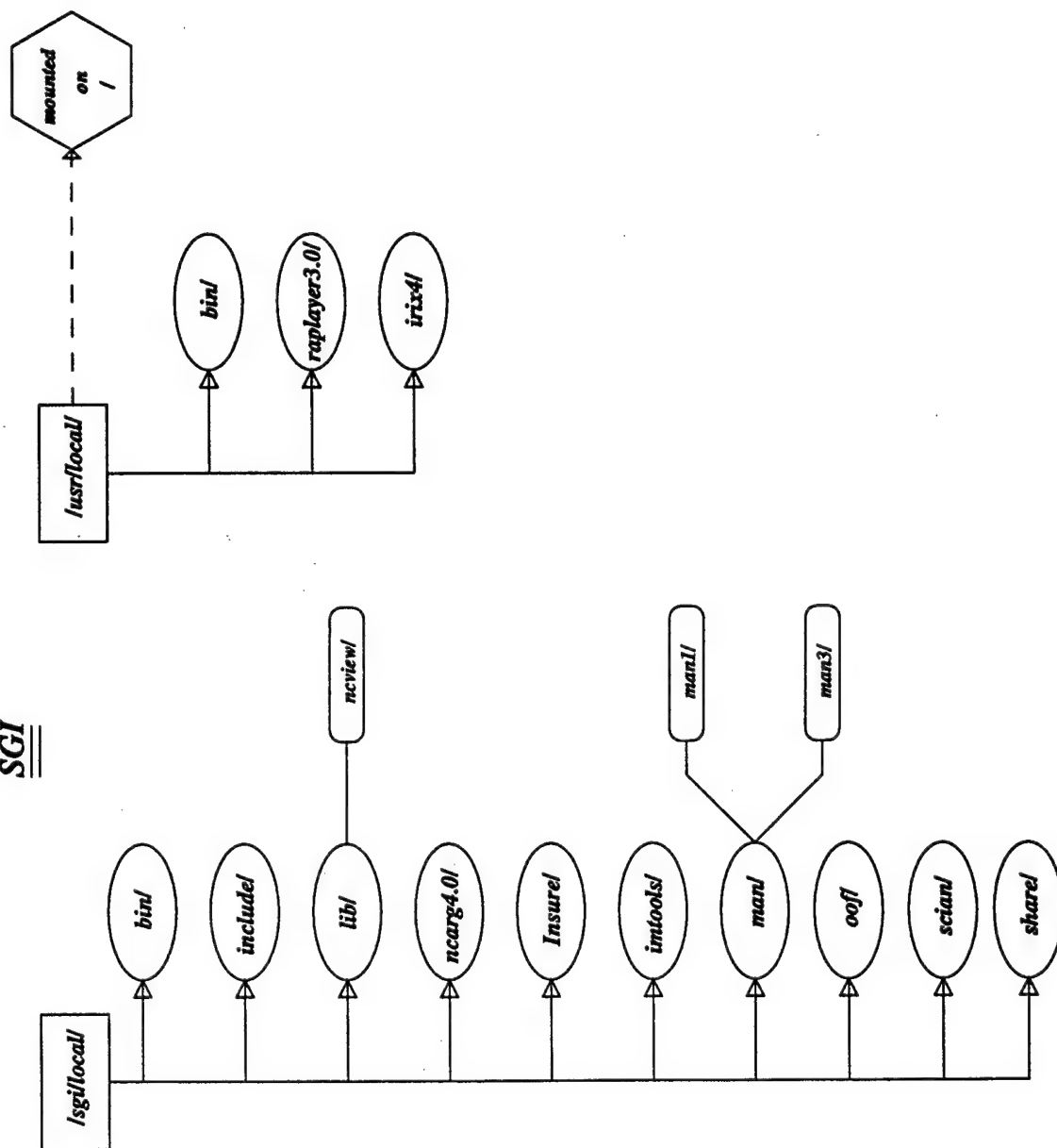


FIGURE 3: SGI Files

**Project #4 Title:** Record Management Tests for the Master Environmental Library (MEL) Project.

**Objective:** To test the record management software for MEL, evaluate the results for various inputs and event procedures via the World Wide Web, and generate a report detailing the errors incurred and possible solutions.

**Approach:** The first step was to become familiar with the software. After noting the various software capabilities, a systematic approach to the tests was made. Notes on program flaws such as error handlers, incorrect executions and processing of data, and invalid relations between the varying segments of this software were made.

**Results:** The generated report specified the approach, and the results of the submission of data or event request in the software. It was also divided to differentiate between the various code parts.

**Further Research Recommendations:** Changes have been made to many software functionalities since these tests were accomplished. Thus, a step through all errors previously noticed should be done.

**Research Advisor:** Mr. Ming Zhou, Center for Air Sea Technology, Mississippi State University.

**Brandon Jockman**  
**B.S. Program in Computer Science**  
**Mississippi State University**

**Project #1 Title:** Conversion Application for Use with the Relocatable Modeling Environment (RME).

**Background:** Global bathymetry data files currently exist in binary format. The CURReNTSS model developed at the University of Colorado contains a binary interpolation algorithm, which requires that these binary files be converted to ASCII format for use in the RME grid generator module, which uses a modified version of this algorithm.

**Objectives:** To locate an existing algorithm for binary to ASCII conversion, redesign it to perform the required processing, utilize it as basis for a stand-alone application, and confirm its accuracy on a new conversion application on existing files.

**Approach:** The existing algorithm was located on a complex file system as an internal Fortran subprocess of an older data processing application. The vital elements of this algorithm had to be extracted and redesigned. Redesigning issues included the conversion from former to current application specifics, dealing with processing bathymetry scale files, accounting for terminal characteristics of both the hardware and programming language in use, and designing the application.

**Results:** The systematic segmentation of data into regional data packets was required to meet Fortran software constraints. The segmentation was performed such that the converted packets were reassembled into a whole. The Fortran conversion application was completed and the simulation results were accurate to the degree of determinability. Any minor data imperfections appeared to be due to the original data and not the conversion process.

**Research Advisor :** Mr. Shuvo Brahmachari, Center for Air Sea Technology, Mississippi State University.

.....  
**Project #2 Title:** World Wide Web Development Utilizing Hypertext Markup Language (HTML), Common Gateway Interface (CGI), and Practical Extraction and Report Language (PERL).

**Background:** HTML, CGI, and PERL are the primary tools used to develop the Master Environmental Library (MEL) Software Version 1.1. I was assigned to assist in the transition and optimization of code for Software Version 2.0. The tools used were PERL Version 5.004, Standard CGI along with CGI.pm PERL module, and HTML Version 3.0.

**Objective:** To become proficient at programming in PERL, HTML, and CGI, and to use these skills to improve the CAST Website.



**Approach:** Once an understanding of PERL syntactics and semantics was obtained, the interaction of PERL with CGI programming was examined. To develop production quality HTML skills, the CAST Webpage was updated. Improvements included the redesign of several page layouts, repair of ImageMaps, and updating/maintaining information stored on the website. A CGI interactive web page featuring technical staff skills was designed to: (1) serve as a dynamic interface between multiple users and the website, (2) consolidate embedding HTML inside the CGI, (3) implement features such as file locking to prevent data corruption from simultaneous multiple user access, and (4) manipulate the format of the user input for storage.

**Results:** The CAST website was improved, and PERL, HTML, and CGI applications were learned.

**Research Advisor:** Mr. Steve W. Payne, Center for Air Sea Technology, Mississippi State University.

.....  
**Project #3 Title:** Design of Communications Between Multiple Programming Languages for the Master Environmental Library (MEL).

**Background:** While nearly all code for the MEL Resource Site Software (MRSS) Version 2.0 was written in the Java programming language, there was code in other languages that was also desired for use.

**Objective:** To design communications between the C++, Perl, and Java programming languages.

**Approach:** While a need existed for embedding Perl subroutines within C++ programs, little documentation dealing with embedding Perl interpreters into plain C was found. Thus a four-stage development plan was used as a proof of concept. First, existing processes for interpreter embedding were observed along with the differences between C and C++. Second, the communication methods were designed based on embedding a Perl interpreter inside C. Third, the communication methods were transferred from C to C++. Finally, practical implementation of this communication upon existing MRSS code was performed.

The task was to develop communication wrappers around vital Perl subroutines for use via pipes in Java. After Java was learned, executable wrappers were written for selected Perl subroutines. Several Java applications were also written to test the designed wrappers. Also, retrofit function prototyping was implemented with Perl to increase the efficiency and accuracy of error detection.

**Results:** To pass parameters to and from C++/Perl, several existing techniques for communication between C and Perl were meshed and then converted to C++ from C. The techniques consisted of perlxs, perlapi, perlgluts, and perlembed. The final mesh consisted of using a Perl interpreter within C++ programs to call existing Perl methods, and manipulate the perlstack with techniques based upon the use of xs language elements to handle return values to the calling C++ code. The primary language differences handled in converting from C to C++ were syntactics and the management of memory.

The communication wrappers designs were straightforward. Java calls the methods via pipes and the parameters are passed via the command line in subsequent elements of the executable call. The return values are returned via standard input. The prototyping method should aid in increasing the efficiency of error detection by forcing a preset format of parameters in Perl methods.

**Future Research Recommendations:** Use the designed proof of concept for development of MEL software, should a conversion to C++ occur in the future. Because the learning curve for basic operations is complex, to be of optimum use, a library of standard functions should be developed to provide an interface for programmers. The Perl wrappers should be bypassed in favor of rewriting the methods in Java to increase the efficiency of execution.

**Research Advisor:** Mr. John Chambless, Center for Air Sea Technology, Mississippi State University.

.....

**Project #4 Title:** Analysis of XML Technology.

**Objective:** To learn the eXtended Markup Language (XML), and analyze this new extension of HTML for usefulness in future CAST projects.

**Approach:** Initially, the World Wide Web Consortium's 10 February 1998 XML recommendation document was studied to obtain information on XML. Several book and online documents were also read, and numerous applications including XML parsers were analyzed.

**Results:** XML is a subset of SGML that was designed to eliminate the complexity, which had prevented industry-wide acceptance of SGML. XML works in unison with both the HTML and SGML markup languages. XML's primary purpose is to allow for transfer and processing of data over the internet with the simplicity of HTML. It also provides a universal storage format and logical data structure. On the receiving end of data transmission, XML processors discern the structure of the data, parse it, and send it to awaiting applications. The capabilities of XML, coupled with the ease of use and interoperability with existing markup, should prove a useful internet tool.

Regarding the integration of XML into future CAST projects, it was noted that a significant amount of effort is required to create well-formed XML structure that will remain useful over several years duration. It was also noted that several aspects of XML, dealing with the current usage of Cascading Style Sheets, are subject to change as the language is still in its early stage.

**Future Research Recommendations:** Future CAST projects would benefit from internet data transmission features. Applications utilizing the XML documents and processors should also be developed. Designs should be structured to keep up with current standards.

**Research Advisor:** Mr. Ming Zhou, Center for Air Sea Technology, Mississippi State University.

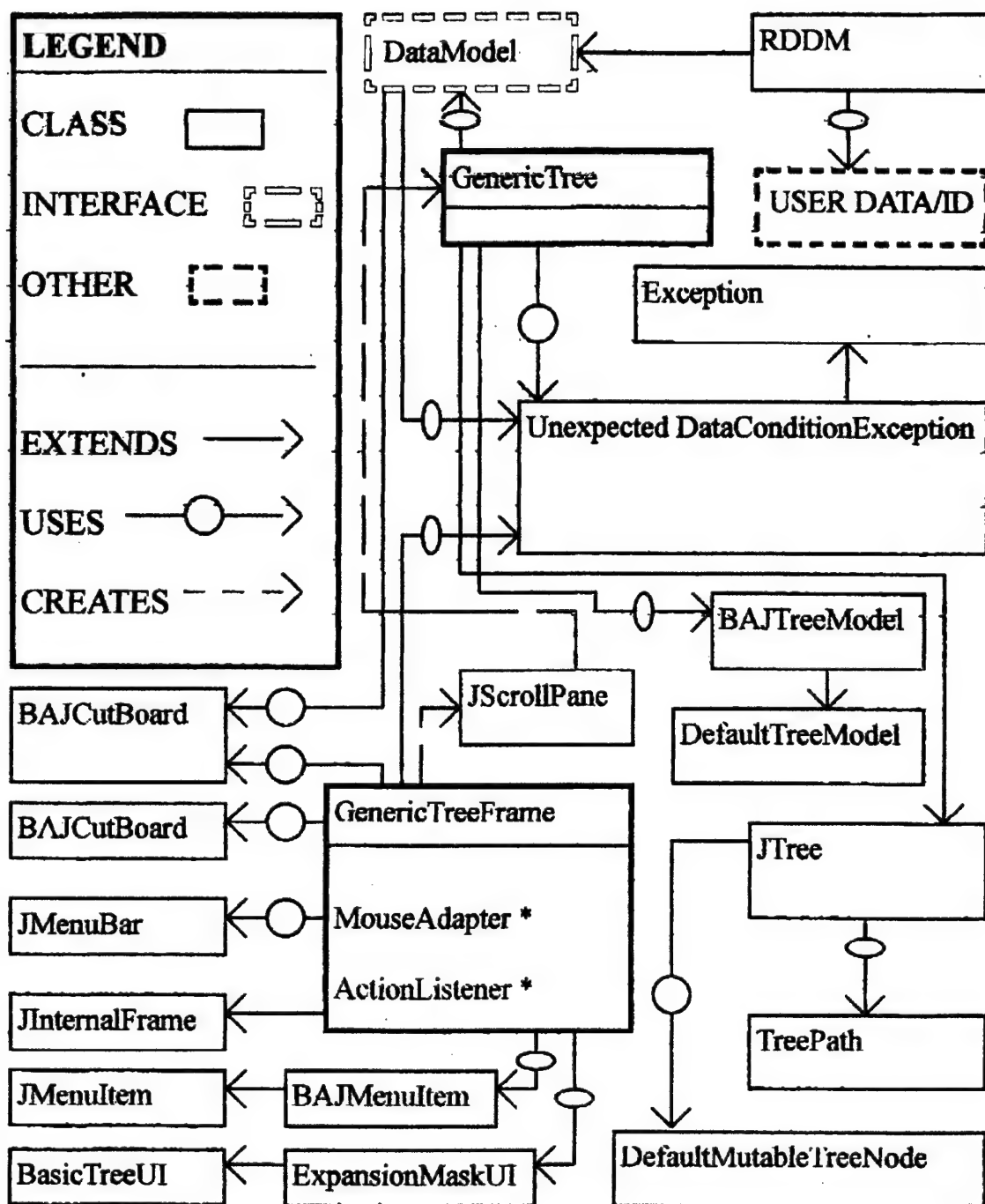
**Project #5 Title:** Embeddable Generic Object Processing Dynamic Tree for Graphical User Interfaces (GUI).

**Objective:** To learn the Java Fundamental Classes (JFC) Swing Toolkit, and utilize it to develop a reusable GUI component tree for future CAST projects. The tree was to be generic in terms of the objects it processes, be dynamic to the extent that individual subtrees can be built at runtime, and be easy to integrate into any windows-based GUI application using Java Technology.

**Approach:** Several JFC resources were analyzed to learn the Swing Toolkit. Sample programs demonstrating its use were obtained. However, all programs found were configured on processing simple character strings as opposed to 'real' objects, and were not helpful as examples of desired behavior. It was decided to experiment with the JTree control and its component classes. Several implementations revealed bugs in the Swing Toolkit Version 1.0, requiring the design of alternative processes, specifically the default mechanism for overriding expansion events.

The GenericTree was designed as shown in Figure 1. Included is the hierarchy of the primary component architecture. All components of the GTree were designed following the same version of the Model-View-Control (MVC) format adhered to by the Toolkit. An interface was designed to allow the GTree to display any type of hierarchical data. This DataModel interface contains methods that must be implemented to successfully connect any data structure to the GTree. The control of hierarchical manipulation is left to the programmer's discretion, although guidelines are well documented in the interface. An example of the DataModel interface was developed and included in the default data model. Labeled as Runtime Dynamic Data Model (RDDM), it uses a hierarchy of employee identification objects ranging from District Manager to Regular Employee to display how well the GenericTree responds to dynamically changing data. Also accompanying the DataModel is an exception class utilized by the interface for unexpected data condition exceptions. This subclass of the exception class is used for returning the status on any future data conditions. The class may be extended to be data type specific for more functionality. By default, the JTree class (the superclass of the GTree) allows objects to be represented only once per tree. This would have presented a problem for some implementations. Consider a file system browser that could contain the same file in separate directories. The problem was solved with the help of a cut and paste board (BAJCutBoard). The DataModel method that corresponds to inserting a node into a new location was given a return value of any java.lang.Object (or any subclass). The addition of one level of abstraction leads to the solution. Instead of having an object placed in multiple locations, outer shell objects should be designed to have member data, which is the object. When the DataModel is instructed to insert a shell object, the implementation will extract the member data object and place it into a new object at that location. Thus the object resides at two positions, but the copy of the reference to the object, that Java is looking for in multiple locations, will not be the same as the differing shell objects at those locations.

GenericTreeFrame (GTF), an extension of the JInternalFrame class, was chosen as the default internal window for displaying the GenericTree in any Java window-based GUI application. The GTF was designed to perform the generation of GTrees of data as well as visual controls of the GTree. The GTF creates a JScrollPane in its content pane. A GenericTree is then created and the visual aspects placed in that scroll pane. The GTF controls all events using listeners and

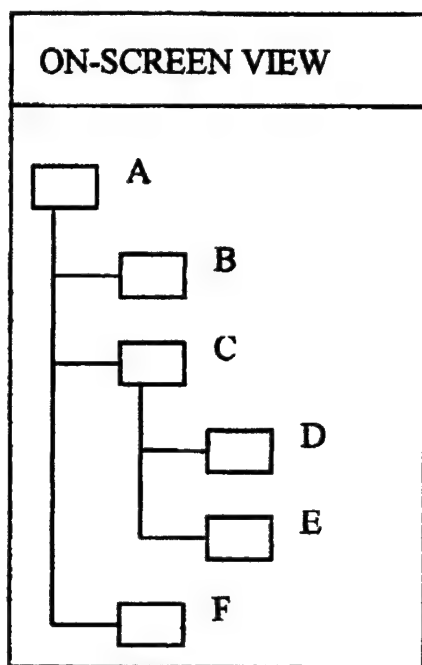


**FIGURE 1: Hierarchy Architecture Of Generic Tree Components**

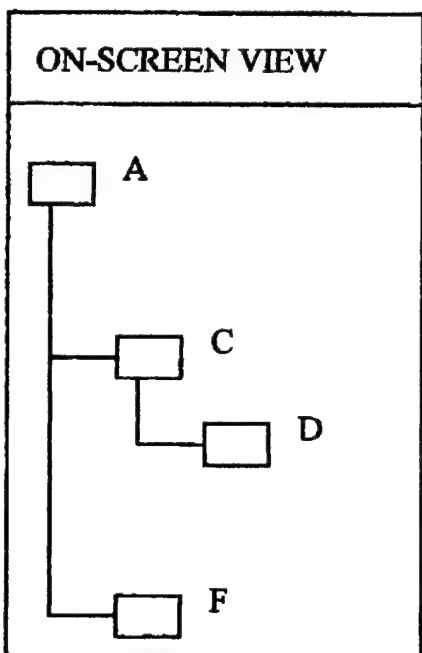
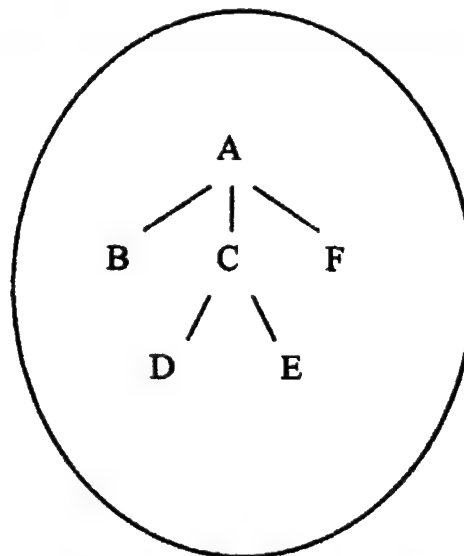
adapters, as well as having separate popup menus for leaf and non-leaf nodes, and a menubar with the ability to refresh the view of the tree displayed. Several tree-utilizing applications were observed to help determine what the default features should be. The default features for the leaf nodes were items for cut, copy, delete, and rename operations. The default features developed for the non-leaf nodes included expand, cut, copy, paste, delete, and rename. These features were chosen as the default popup menu sets for their ability to perform widely used operations for any type of data. Several other features were considered and ruled out as being type-specific operations and therefore limiting the objects the GTree could process. To aid in the cut/paste operations of the popup menus, the BAJCutBoard class was created. Cut and copied objects are stored in the clipboard emulator. As discussed earlier, a problem occurs when the same object is placed in multiple locations in the tree. This occurs with pasting an object in the clipboard if it already resides in the tree. To correct this, the DataModel interface's object insertion method returns an optional object. This allows the interface implementers to check the data hierarchy for the existence of this object and return a new object corresponding to its newly inserted location for the GTree to use. Thus, objects existing in the trees can be pasted into multiple locations. The implementation of the refresh menubar item was designed to maintain the integrity of the existing visual hierarchy. When a node is selected, GTW confirms that the node is still valid. If not, indication is given that the tree should be refreshed. Then, the runtime data content state is saved and restored to the point of validity. This is shown in Figure 2. Regarding the default expansion event handling bug, the BasicTreeUI was extended to ExpansionMaskUI, which overrides the drawing of the expansion, handles on screen, and also handles clicks. The mouseListener in GTree was implemented to cause the expansion/collapsing of nodes when individual nodes are clicked. The default renaming ability was then adjusted to corresponding menu item events to avoid conflicts with the new expansion process. This changed the node expansion and bypassed the code of the default handler.

The GenericTree itself (an extension of JTree) was designed to perform the actual tree creation and DataModel interaction (except for the cut/paste mentioned earlier). It was designed to utilize both the DataModel interface and a BAJTreeModel that extends DefaultTreeModel to maintain the new renaming process. The reason for using two separate data models is that the individual subtrees rooted at non-leaf nodes of the GTree are built at runtime. When the user clicks on a node, the contents are located through the DataModel interface and into the BAJTreeModel which contains the currently known Gtree nodes. This allows the GTree to refresh operation of its frame's menu to maintain the integrity of the data currently loaded, while not constraining the tree by having a preset model that would have to be loaded in its entirety every time the data was changed in other locations. DefaultMutableTreeNode and TreePath were used to house the objects being processed and to locate nodes corresponding to on-screen clicks. The processes of both GenericTree and GTF were designed to be non-data type specific. Thus, any object that is an extension of java.lang.Object is a valid object for usage with GTree.

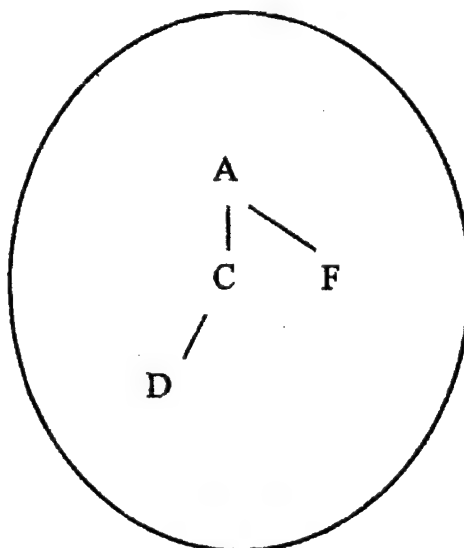
**Results:** Of the bugs located in the Swing JTree and accompanying component classes, the most serious caused the JTree to lose track of the visual representation of currently expanded nodes. This was due to the visual aspects of the tree being handled prior to the default mechanism, providing for the insertion of additional functionality in the handling of expansion events. A new process for expansion handling was developed to bypass this problem.



**DATA NODES INITIALLY IN  
DATA MODEL**



**AFTER DATA NODES B AND E  
ARE REMOVED FROM MODEL.**



**FIGURE 2: Relative Views Of Onscreen/Data Model**

The fundamental/top-level components developed were the GenericTree, GenTreeFrame, and DataModel classes. To integrate the GTree into any Java developed window GUI, the programmer needs only to implement the DataModel interface and create/include an instance of GTF inside a valid outer window frame. A workshop to demonstrate the default features of GTree was given. The GenericTreeWorks(GTW) was also used to test the features of the GTree in a variety of situations. It also demonstrated the use of GTF's menu and popup menu item operations with parent window menu selections. The creation of multiple Gtrees, referring to the same data model, demonstrated how data may be shared. When data was changed using one window, appropriate changes were reflected in the other windows sharing the data.

**Future Research Recommendations:** Research is continuing regarding implementation in CAST projects. The design of several useful components that include transitioning the GTW operations to a browser-like splitpane and allowing objects to carry their own icons for file folder on-screen representations are in progress.

**Research Advisor:** Mr. Ming Zhou, Center for Air Sea Technology, Mississippi State University.

.....

**Brandon Jockman**  
**B.S. Program, Computer Science**  
**Mississippi State University**

**Sayantana Ganguly**  
**B.S. Program, Computer Science**  
**University of Southern Mississippi**

**Project #6 Title:** Alpha Testing and Code Revisions for the Master Environmental Library (MEL) Resource Site Software (MRSS).

**Objective:** To perform tests on MRSS components with the focus towards new features and functionality of the Administration Tool and the Data Authorization Mechanism.

**Approach:** The initial phase consisted of studying the architecture of MEL. Several MRSS installations were to be installed on several different UNIX configurations including Sun and SGI workstations. A format for bug logs was devised with areas assigned for name, location, description, date, and fix. Then, a systematic testing of the software would be performed to include a code review, wherein selected segments were rewritten to eliminate unnecessary code.

**Results:** Logs were assembled for both software and documentation. All problem areas were either repaired/eliminated and the code optimized.

**Research Advisor:** Mr. Sachin Bhate and Mr. John Chambless, Center for Air Sea Technology, Mississippi State University.



**Colin Murray**  
**M.S. Program in Physical Oceanography**  
**Florida State University**

**Project Title:** A Three Dimensional Scalar Field Initialization Program for the Baroclinic Tidal Model in the Relocatable Modeling Environment (RME) Project.

**Background:** The goal of the RME project is to increase the Navy's ocean prediction capability. The RME includes a GUI system for setting up and running numerical ocean models. It allows for convenient user specification of the model domain and ancillary datasets such as bathymetry and initialization climatologies. The software described herein is part of an effort to incorporate the data-assimilating baroclinic tidal model, developed by Dr. Lakshmi Kantha of the University of Colorado, into the RME.

**Objectives:** Ocean models are essentially sets of coupled partial differential equations numerically integrated forward in time. As such, they require initial conditions. The baroclinic tidal model requires initial temperature and salinity fields in three dimensions corresponding to its sigma coordinate model grid. The objective was to develop a computer program to map scalar fields from the Levitus Climatology (World Ocean Atlas of 1994, NODC, Washington, DC) onto an arbitrary three-dimensional sigma-coordinate model grid. Two major design requirements had to be met. First, every model ocean grid point must be assigned a valid value. Second, because the program was to be integrated into the RME, it has to accommodate input arrays of arbitrary size.

**Approach:** The input was specified as three like-sized two-dimensional arrays containing bathymetry, longitude, and latitude values of the model domain; two integer variables containing the dimensions of these arrays; an integer variable containing the number of sigma levels to be included in the output array; and a one-dimensional array containing depth conversion ratios for the sigma levels in the output array. The output was to consist of two like-sized three-dimensional arrays of temperature and salinity values. The two horizontal dimensions must match the bathymetry grid while the vertical dimension must match the number of sigma levels.

The overall software architecture would consist of several major steps. First, the global Levitus databases of temperature and salinity were read in from external files. The datasets consisted of one degree by one degree latitude-longitude values in the horizontal and thirty-three unevenly spaced z-levels in the vertical. The z-levels were expanded to give higher resolution near the surface where gradients were highest. Using the model grid longitude and latitude information, subsets horizontally circumscribing the model bathymetry but containing all thirty-three z-levels were extracted.

Second, each Levitus z-level in the subsets was interpolated onto the horizontal model grid. The interpolation was carried out using a generalized linear scheme where each immediately surrounding valid climatology value contributes in proportion to the inverse of its distance from the interpolated point, normalized by the sum of the inverse distances of all contributing climatology values. If the four climatology values immediately surrounding a model ocean grid point were invalid, where invalid means either land or a missing ocean value, then 'boxes' of



values successively farther away were tested until a valid data value could be assigned. The resulting intermediate temperature and salinity data arrays had horizontal dimensions matching the bathymetry and a vertical dimension matching the thirty-three z-levels of the Levitus database.

These intermediate arrays were then checked to ensure that all model grid points had valid data values. This step was necessary because problems could arise due to inconsistencies between the input bathymetry and the implicit bathymetry of the Levitus climatology. The first z-level was always reliable, so each successively lower level was checked and any problem points were filled either by linear interpolation if between two valid values or assigned the data value of the point immediately above it. Finally, each column of the intermediate data arrays was mapped from z-level to sigma-level coordinates. The sigma levels for a given column were found by multiplying the depth at that gridpoint by the sigma ratios. The resulting value in meters was the reference location in the water column to which the z-level values were mapped. The surface sigma level and any sigma levels deeper than the deepest valid z-level were found using linear extrapolation of the two nearest values. In very shallow water where only one valid z-level exists, all sigma-levels were assigned the value. Any sigma level that fell between two valid z-levels was assigned a value by linear interpolation. The resulting three-dimensional temperature and salinity arrays corresponded exactly to the model sigma-level grid and were output to the RME for model initialization.

**Results:** The project resulted in the completion of a robust scalar initialization program, which was successfully integrated into the RME.

**Research Advisor:** Mr. Shuvobroto Brahmachari, Center for Air-Sea Technology, Mississippi State University.

**Shengyong Li**  
**M.S. Program in Computer Science**  
**University of Southern Mississippi**

**Project #1 Title:** Research the Oracle8 Database Management System and Web Server System Configuration.

**Background:** Due to the frequent name changing among Oracle products, and the inconsistency of the product names in Oracle's marketing materials, it is difficult to understand the functionality of some Oracle products.

**Objectives:** To identify various Oracle8 software components and their features, to identify the hardware/software configuration requirements for the Oracle8 installation, and install Oracle8/Oracle Web Server on a NT 4.0 workstation.

**Results:** This research clarified the differences between various Oracle software components, and helped CAST decide which products to use. A prototype installation of Oracle DBMS/Oracle Web Server was also performed.

**Research Advisor:** Mr. Ming Zhou, Center for Air Sea Technology, Mississippi State University.

.....  
**Project #2 Title:** Development of an Extractor for the Master Environmental Library (MEL).

**Objectives:** To develop software to enable collection of user-specified data for delivery via MEL.

**Approach:** The target data set was the Advanced Three-Dimensional Circulation (ADCIRC) Model which produces tidal elevation and current components at one hour increments. Data was available for fifteen locations near Camp Lejeune, North Carolina, in March 1997.

An extractor was required to analyze the user's data requirements and obtain user's choices for parameter, time, and locations. Since MEL Regional Site Software is written using Perl Version 5.003, the extractor had to be written in that language, or a Perl interface had to be written that could call an extractor written in another language.

The extractor final design consisted of a Perl interface, which launched a stand alone extraction application written in C. The Perl interface code identified the parameters, locations, and time range. These were passed as arguments to the C extractor. The C application then searched the flat file database for appropriate dates within the time range, opened the correct file, and read the requested parameters for the desired locations. A separate file for each date was created for the parameters at all requested locations. The C application then terminated and returned control to the Perl interface. Finally, the Perl interface passed control back to the Resource Site Software, which packaged the data for delivery to the customer.

**Results:** The ADCIRC extractor is presently in use within the MEL Resource Site Software installation. In addition, the original ADCIRC extractor has been used as the basis for other extractors with similar requirements.

**Research Advisor:** Mr. Steve Foster, Center for Air Sea Technology, Mississippi State University.

.....  
**Project #3 Title:** Development of an Extraction Library Module for the Master Environmental Library (MEL).

**Objectives:** To analyze extractors being used by the MEL Regional Resource Site Software, and thus identify functions and algorithms superficially common to several extractors. The goal was to create a library module for these functions and algorithms written in the Perl language.

**Approach:** A survey of existing extractors identified six blocks of code (either Perl subroutines or algorithms) that could be placed in a modular library for common use. Each candidate block of code was re-designed for implementation as a Perl library module subroutine that could be called by an external Perl routine. The six subroutines implemented were:

- Write comments to a standard "README" file;
- Identify incremental times at stepped intervals within a time range;
- Perform date/time calculations (addition, subtraction, multiplication and division);
- Determine if a spatial location is within a bounded area;
- Launch a process to perform data encoding using the Binary Universal Form for the Representation (BUFR) of meteorological data; and
- Write a directory listing to a file.

Once the library module was designed and tested, all extractors were modified for use.

**Results:** The Perl extraction library module is now being used as a standard element of the CAST data extraction effort for MEL.

**Research Advisor:** Mr. Steve Foster, Center for Air Sea Technology, Mississippi State University.

**Mike H. Zhou**  
**Ph.D. Program in Scientific Computing**  
**University of Southern Mississippi**

**Project #1 Title:** Development of a Virtual Data Warehouse (VDW) for the Environmental Protection Agency/Gulf of Mexico Program Office (GMPO).

**Objectives:** To develop a VDW that provides a World Wide Web interface to the EPA/GMPO meta-data database.

**Approach:** The VDW consists of a client GUI, web server, and meta-data database. The system is in a PC Windows NT server environment. On the server side, it uses active server pages (ASP) technology, and is more efficient and easier to use than the CGI interface of the ISAPI (Internet Server Application Programming Interface). The ASP is also more powerful and flexible than dbweb. For the database, Microsoft's SQL 6.5 was used. This database engine was comparable to ORACLE. ASP interfaced the database through the Open Database Connection (ODBC) on the NT. ODBC and could be connected with other database engines such as MS Access.

On the client side, the JavaScript Language was used to provide a GUI for interaction between the user and the browser. This resulted in a user-friendly and fast GUI, since much of the work was done in the browser, with little communication overhead between the browser and the server.

**Results:** Users can now insert, update, and retrieve information in the database through a web browser.

**Future Research Recommendations:** Use Java applet to provide a better GUI.

**Research Advisor:** Mr. Ming Zhou, Center for Air Sea Technology, Mississippi State University.

.....  
**Project #2 Title:** Development of the Enterprise Information Explorer.

**Objectives:** The Enterprise Information Explorer (EIE) system helps enterprise employees inventory and explore corporate information, and data not necessarily in the form of database tables. The files can be most anything including MS word, spreadsheets, web pages, movies, etc.

**Approach:** The entire system is in a PC Windows NT Server environment. While the system will not move the files of individual employees to a centralized place, a meta-data database is built to keep track of and organize the files. The information on each individual file is stored as records of the objects table. The "relations" are used to help the user find all the files related to a found file. Examples are "has employee" and "has projects". The "Associations" associate the objects through relations.

**Rohit Mehra**  
**B.S. Program in Computer Science**  
**University of Southern Mississippi**

**Project #1 Title:** Enhancement of the Naval Interactive Data Analysis System (NIDAS) in Axes Units.

**Objective:** Provide a NIDAS user the ability to specify units of depth, temperature, and sound speed. Provide appropriate labels along the axis.

**Approach:** The depth units (meters and feet) already existed in NIDAS Version 4.0. The functions for fathom, centigrade, and fahrenheit were to be added. A separate routine would be needed to keep up with the change in units and to provide appropriate axis labels. A label would also be needed to display the user's current settings.

**Result:** A separate routine called changeAxis was created to track the change in units. A button was created to provide the user with an easy way of changing the units and to display the current setting. The specified units remain unchanged, when changing from temperature to salinity or to any other parameter, or when switching to multi-view, where there were six graphical windows with each showing a different parameter. When the unit was changed in one graphical window within multiview, the associated units in the other windows were also changed. This provided the user with unit consistency over all graphical windows. When switching back to the default values, the units were changed back to meter and centigrade. These various functions were successfully tested and integrated into NIDAS.

**Research Advisor:** Mr. Clifton Abbott, Center for Air Sea Technology, Mississippi State University

.....

**Project #2 Title:** Implementing a Data Category Interface.

**Objective:** The Motif/X-Window environment is used to implement many of the applications developed at CAST, and was to be used to implement a data category interface for CAST data.

**Approach:** To first learn Motif/X-Windows programming through the use of manuals and sample programs. After developing the needed skills, design and implement an interface to allow the user to select the category their data best fits.

Attended a three week programming class on Motif/X-Windows presented by Mr. Clifton Abbott. The concepts discussed were event handling, widget types, widget creation, managing widgets within the application, and compiling using "Makefiles". Several sample programs were implemented to enhance Motif skills.

**Results:** The data category interface was implemented to provide the user with several toggle buttons to choose which category best fit the data. The data categories were located in an

external directory, and the user can have only one category selected at the same time. When the user is finished, the category is saved.

**Research Advisor:** Mr. Clifton Abbott, Center for Air Sea Technology, Mississippi State University.

.....  
**Project #3 Title:** Design of a NetCDF File Format for the Comprehensive Oceanographic and Atmospheric Data Set (COADS).

**Objective:** Cast handles COADS data in support of NIDAS, and a NetCDF file format was to be designed to store COADS metadata.

**Results:** After learning NetCDF, developed skills in its use through studying manuals and by writing sample programs to create files and add data. A format was designed to store the minimum and maximum values for winds, waves, cloud cover, fog, precipitation, and air temperature.

**Research Advisor:** Mr. Clifton Abbott, Center for Air Sea Technology, Mississippi State University.

**Min Zhang**  
**Ph.D. Program in Biological Science**  
**University of Southern Mississippi**

**Project Title:** The Application of JavaScript Technology to the EPA/GMPO's Virtual Data Warehouse (VDW) Graphical User Interface (GUI)

**Objective:** To learn JavaScript and apply it to the VDW GUI.

**Approach:** The JavaScript language was developed by Netscape Corporation. It is a programming language that allows scripting of events, objects and actions to create an internet application, and has the characteristics of:

- An object-oriented programming language. JavaScript enable browsers are modeled into a browser's object model. Thus, every element and tags on HTML pages are loaded into the object model so that each element has its own methods and properties, and can receive its own events. The object model is hierarchically organized into a tree-like structure. JavaScript interacts with the web browsers through the browser's object model, and allows one to define his own object through new statements.
- An embedded and script language. It is not a general-purpose object-oriented programming language. It is designed to provide a quick and simple method to enhance web pages. JavaScript is embedded into HTML through a script tag and is then downloaded with an HTML file into the browser, where it is interpreted and executed.
- An event driven language. Elements of a web page, such as a button or checkbox, may trigger an event. When this occurs, a corresponding piece of JavaScript code, usually a function, is executed. This function then performs algorithms that modify the contents of the web pages or validate the information.
- Syntactically similar to C++ or the Java programming language.

The approach used was to design the VDW front end (Web pages) from the view of a database application. An authorized user may retrieve, update, insert, and delete records from the database. Thus, the GUI is designed into different modes such as query, creating, and modifying. The record structure of the database is graphically represented as a folder directory tree structure that is dynamically changed to correspond to the record status. The data records are actually accessed through forms located at the bottom frame of the web page, and are saved into a JavaScript array object stored in the client site machine memory. Thus, the data can be validated and modified at the client site without interference from the server, and data transmission traffic on the network is reduced. Other functions such as validating users, sending comments, and sending parsed FGDC format data into the VDW format back to client, need JavaScript technology to tie the different technologies together.

**Results:** The implementation of the VDW GUI with HTML and JavaScript was successful. All functions such as representing the record tree structure dynamically, changing modes, displaying

the whole record in different windows, and coordinating different contents in different frames were achieved. When other technologies such as the Microsoft Server Side Active Server Pages are combined with JavaScript, they will produce impressive and dynamic web pages. However, some problems can occur. For example, since JavaScript is executed at the client side web browser, it is difficult to predict the behavior of JavaScript on different platforms on which the browser is running, as well as on different browsers and different versions of the same browser. This problem can be alleviated with more browsers to support JavaScript and with unified JavaScript standards. Another problem occurs when JavaScript has been running for a long time, and this sometimes causes the browser to behave erratically. The cause is not known.

**Research Advisor:** Mr. Ming Zhou, Center for Air Sea Technology, Mississippi State University.



## **DISTRIBUTION LIST**

1. Office of Naval Research  
800 N. Quincy Street  
Arlington, VA 22217-5000
  - Dr. Emanuel Fiadeiro (3)
  - Mr. William McCarthy (Code 00CCI)
2. Administrative Grants Officer  
Office of Naval Research Resident Representative  
101 Marietta Tower-Suite 2805  
101 Marietta Street  
Atlanta, GA 30303
3. Director, Naval Research Laboratory (Code 2627)  
Washington, DC 20375
4. Defense Technical Information Center (2)  
8725 John Kingman Road, Suite 0944  
Fort Belvoir, VA 22060-6218
5. Naval Oceanographic Office  
Stennis Space Center, MS 39529
  - Dr. Martha Head
  - Mr. Steve Haeger
6. Director, Sponsored Programs Administration  
Mississippi State University  
PO Box 6156  
Mississippi State, MS 39762
7. Naval Research Laboratory  
Stennis Space Center, MS 39529
  - Mr. Rick Allard
  - Dr. John Blaha
8. Naval Research Laboratory  
Monterey, CA 93943
  - Dr. Richard Siquig
9. Technical Director  
Commander Naval Meteorology and Oceanography Command  
Stennis Space Center, MS 39529
10. Environmental Protection Agency  
Stennis Space Center, MS 39529
  - Dr. Eugene Meier
  - Mr. James Matthews

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. Agency Use Only (Leave blank).	2. Report Date. September 30, 1998	3. Report Type and Dates Covered. Technical Note 1 Oct 1997 - 30 Sept 1998
-----------------------------------	---------------------------------------	---

4. Title and Subtitle.  FY98 STUDENT RESEARCH PROJECTS	5. Funding Numbers.  Program Element No.  Project No.  Task No.  Accession No.
--	--

6. Author(s). S. Brahmachari, S. Payne, P. Ehrensing, S. Ganguly, Mike Zhou, B. Jockman, J. Chambless, Ming Zhou, S. Bhate, C. Murray, S. Li, S. Foster, G. Murchison, R. Mehra, C. Abbott, M. Zhang, J. Corbin, and L. Yeske.

7. Performing Organization Name(s) and Address(es). Mississippi State University Center for Air Sea Technology Building 1103, Room 233 Stennis Space Center, MS 39529	8. Performing Organization Report Number.  CAST Technical Note 2-98
---	--

9. Sponsoring/Monitoring Agency Name(s) and Address(es). - Office of Naval Research 800 North Quincy St., Arlington, VA 22217 - Naval Research Laboratory, Stennis Space Center, MS 39529 - NASA, Stennis Space Center, MS 39529 - EPA, Stennis Space Center, MS 39529	10. Sponsoring/Monitoring Agency Report Number.  CAST Technical Note 2-98
---	--

11. Supplementary Notes. Research performed under ONR Research Grants N00014-95-1-0203, N00014-97-1-0099, and N00173-98-C-6012; NRL Contracts N00014-95-C-6032 and N00173-98-C-6012; NASA NAS13-564 Delivery Orders 131 and 132; and EPA Grant MXS984100-97-0.

12a. Distribution/Availability Statement.  Approved for public release; distribution is unlimited.	12b. Distribution Code.
--	-------------------------

13. Abstract (Maximum 200 words).  
  
Numerous FY1998 student research projects were sponsored by the Mississippi State University Center for Air Sea Technology. This technical note describes these projects which include research on Graphical User Interfaces, Master Environmental Library, Database Management Systems, Naval Interactive Data Analysis System, Relocatable Modeling Environment, Tidal Models, Book Inventories, System Analysis, World Wide Web Development, Virtual Data Warehouse, Enterprise Information Explorer, Equipment Inventories, COADS, and JavaScript Technology.

14. Subject Terms. (U) Technical Note (U) Student (U) Research (U) CAST (U) ONR (U) NRL (U) NASA (U) EPA			15. Number of Pages. 35
			16. Price Code.
17. Security Classification of Report. UNCLASSIFIED	18. Security Classification of This Page. UNCLASSIFIED	19. Security Classification of Abstract. UNCLASSIFIED	20. Limitation of Abstract.